

SHADOW

Step-by-Step Intrusion
Detection using TCPdump

Objective

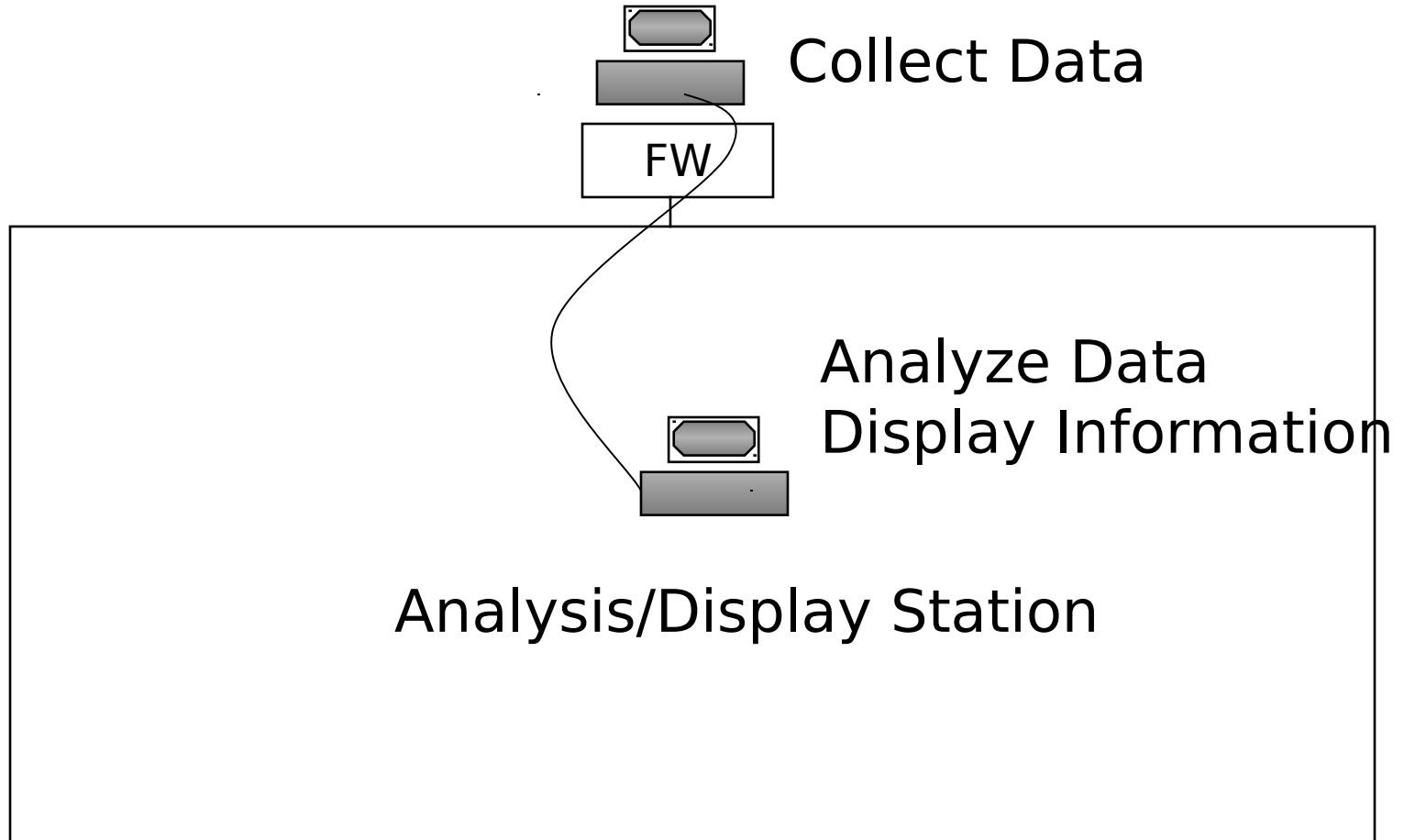
The objective of this project is to familiarize you with the SHADOW documentation and give you the confidence that you can build an intrusion detection system.

<http://www.nswc.navy.mil/ISSEC/CID>

What do I need?

- Unix experience including compiling software
- 2 Unix workstations, PC Pentiums running Linux or Free BSD are preferred
- At least 9 Gigabyte disk per system
- SHADOW software (FREE)

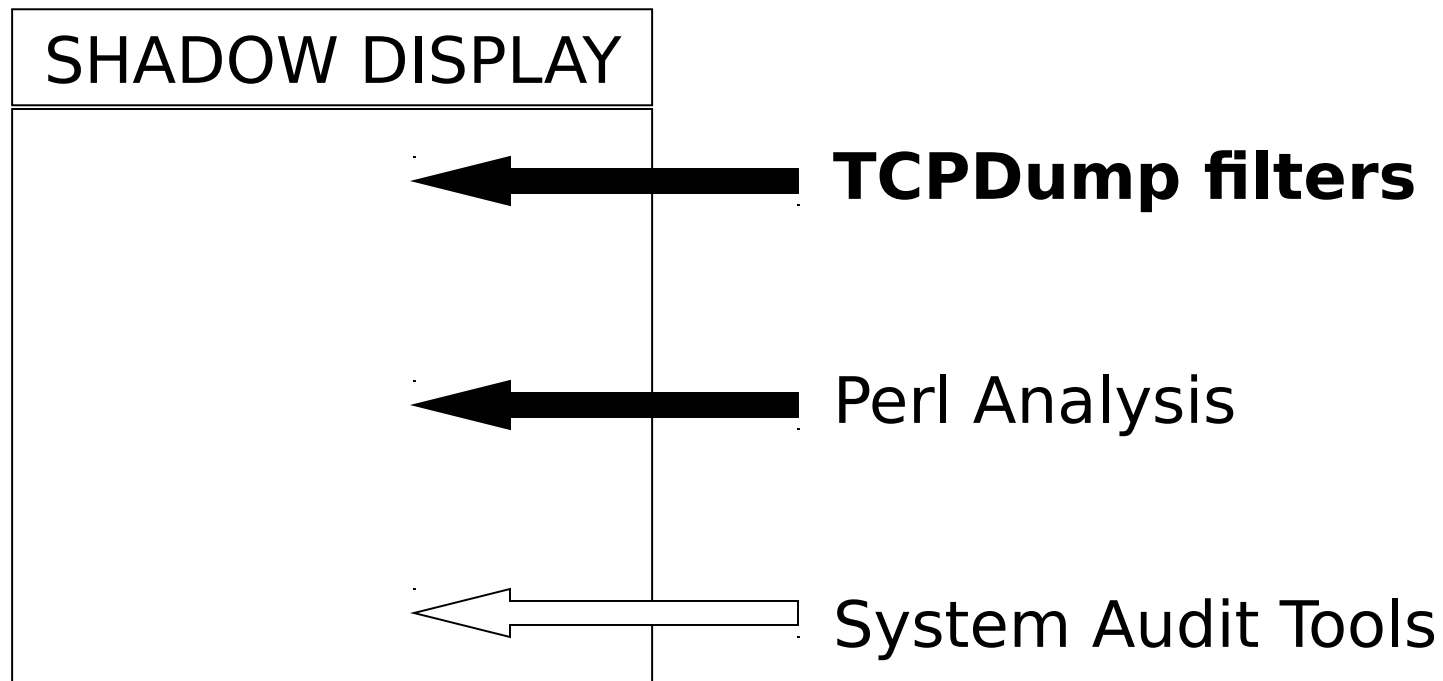
Overview of the Architecture



Architecture as viewed by CIDF

S	Push, produces GIDOs in response to events	Pull, produces GIDOs when queried
A	Receives pushed events	Queries for data

SHADOW as a Framework



If it can display as text for html it can be used

Why TCPdump

- Libpcap
- Compiles on many Unix platforms
- High fidelity
- Same program for data collection and first order analysis

Is this a burglar alarm or a traffic analysis based intrusion detection system?

Proposed ID Architecture

R
e
a
l

T
i
m
e

O
n
e

H
o
u
r

H
i
s
t
o
r
i
c
a
l

Time



Getting the software

We acquired our tcpdump software from:

<ftp://ftp.ee.lbl.gov>

The program will be labeled tcpdump.tar.Z
Make sure you also get libpcap, (libpcap.tar.Z),
since that is how the Unix system gets the
network information from its kernel.

These software packages have been made
available by the Network Research Group at the
Lawrence Berkeley Laboratory.

Build the sensor

- Cron, calls log_driver.pl when it is time for it to do work
- log_driver.pl, sets up variables and calls the other scripts
- stop_logger.pl, stops the sensor so a new file can be started
- start_logger.pl, starts the new file

Build the Analysis Station

- Cron, at appropriate times (every hour) cron calls fetchem
- fetchem, downloads the last hours data and runs tcpdump on the data using the bad_events filter
- filters, are how we extract information from the tcpdump data file

Filters

- Tcpdump is run on data file with filters to print the desired output
- Filters are created from simple primitives and strung together as needed
- Common connectors: and, or , not

IMAP Filter

tcp and dst port 143

NFS Filter and Results

ip and udp port 2049

```
05:17:50.562188 jokull.Colorado.EDU.885592240 > dorado.nswc.navy.mil  
l.nfs: 40 null  
17:52.553265 jokull.Colorado.EDU.885592240 > dorado.nswc.navy.mil  
l.nfs: 40 null
```

Easy Does It

It turns out that it takes some experience to learn to tell “good” packets from “bad” packets. Be slow as you begin your journey into intrusion detection to raise the alarm. Give yourself several weeks to watch your data and learn your organization’s network.

Tuning a Filter

**tcp and (dst port 143) and not
(host.goodguy.org or net 192.168.4)**

Core_Hosts Filter

- DNS, web and mail servers draw a lot of fire, about 20% of all our attacks are directed at these systems
- If you lose control of DNS, they own you
- Worth the time to give connection attempts to these systems an extra look

Core_Host Filter Web Server

```
(dst host 192.168.1.1 and
(
  (tcp and ((tcp[13] & 2 != 0) and (tcp[13] & 0x10 = 0))
    and (not dst port 80))
  or
  (udp and not dst port 53 and not dst port 137)
  or
  (icmp and (icmp[0] != 8) and (icmp[0] != 0)
    and (icmp[0] != 3) and (icmp[0] != 11))
  or
  (not (tcp or udp or icmp))
))
```

Core_Host Filter Web Server

The interpretation

```
# 192.168.1.1 webserver
# should only receive traffic to tcp port 80 (syn only)
#   ignore udp with dst port 53 or 137
#   ignore icmp echo requests (8), echo replies (0),
#   destination unreachable (3), and
#   time exceeded (11) error messages
```

The SHADOW documentation literally has pages of sample filters and explanation!

Bad_Events Filter

- Complex filter that picks up the things you want to watch for
- Be willing to let a little “noise” in
- Scans and noisy probes will be immediately obvious by the file size

Bad_Hosts Filter

- Once you determine that a host has attacked/attempted to attack you, add them to the bad_hosts list
- This is one way partnering with other organizations you trust is a major win
- Remember there is a potential **denial of service** if you block these host/nets.

Closing Note on Filters

We have just hit the high points on filters, the Step-by-Step Intrusion Detection using tcpdump has more examples and discussion. There is no substitute for your trying various filters!

Display

- Output of complex filters such as `bad_events`, `bad_hosts`, `core_hosts` can be written as text files to an intranet web server
- This way various people can share the duty of reviewing the files (easily).
- Before we display we sort the detects by SRC address then resolve names

Display

```
01:53:43.647688 ATHM-209-218-xxx-2.Home > 147.168.255.255: icmp: echo request
01:53:44.049125 ATHM-209-218-xxx-2.Home > 147.168.0.0: icmp: echo request
01:53:44.649461 ATHM-209-218-xxx-2.Home > 147.168.255.255: icmp: echo request
01:53:45.079945 ATHM-209-218-xxx-2.Home > 147.168.0.0: icmp: echo request
```

num	dests	source ip	source name
9		256.172.1.43	venus.srn.edu
5		256.0.14.129	k.root-servers.net
5		256.41.0.21	srrn-servers.net
46		256.93.1.190	we.were.bombed.at.empact.or.jp
10		256.115.155.132	tnt1.srn.ca.da.uu.net
272		256.147.90.21	
30		256.115.125.201	madcrew.srn.org

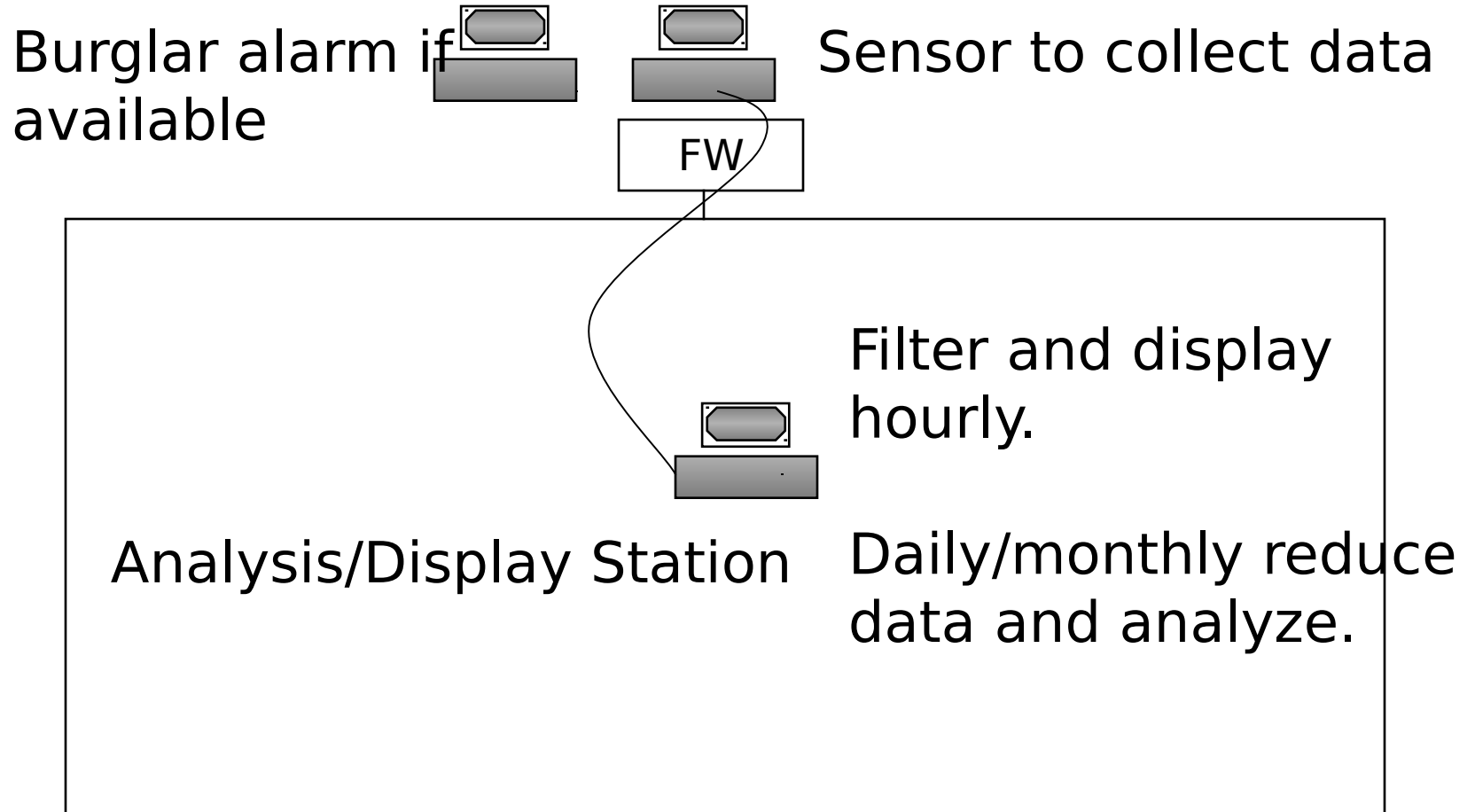
The 80 - 20 Rule

- It is generally possible to achieve 80% of the results with 20% of the effort. We are now at this point.
- You should have a working ID system that reduces data about (possible) attacks and keeps several days of full data online

Going for the other 20

- Use system in concert with burglar alarms which detect in near real time
- Hourly analysis of data to detect scans that filter matching missed, use this information to tune filters
- Correlation with regional class system if available

Implementing the Model



Data Reduction

- Tcpslice to “cat” files to a 24 hour day
- Use filters to separate traffic by protocol tcp, udp, icmp, routing and other (tcpdump -r file udp) etc
- Consider how you want to reduce tcp, are SYN packets sufficient?

Extra Credit

- Add the reduced data to the “sliding window”
- Resolve all addresses against the “big host table”
- Compare activities of hosts against their profiles
- Flag the mismatches, attack patterns, etc

Summary: You can do it!

- Unix experience including compiling software
- 2 Unix workstations, PC Pentiums running Linux or Free BSD are preferred
- At least 9 Gigabyte disk per system
- SHADOW software (FREE)